IN THE CLAIMS

Please amend the claims as follows:

1. (Currently Amended) A method, comprising:

dynamically establishing a first debugging session with a first processing core of a processor;

dynamically establishing a second debugging session with a second processing core of the same processor; and

concurrently managing the first and second debugging sessions independently from one another on the same processor, wherein the first processor is a multi-core processor architecture that supports the first processing core and the second processing core, and wherein the first debugging session operates independent of the other debugging session and both sessions independently and simultaneously proceed within the same processor on their respective processor cores, and the first debugging session is associated with a first application and first user that is different from the second debugging session that is associated with a second application and a second user.

2. (Original) The method of claim 1, further comprising:

initiating a first instance of a debugger on the first processing core of the processor for interactions occurring during the first debugging session; and

initiating a second instance of the debugger on the second processing core of the processor for interactions occurring during the second debugging session.

3. (Original) The method of claim 1, wherein dynamically establishing the first and second debugging sessions further comprises dynamically establishing the sessions by connecting the first and second processing cores of the processor to separate instances of a debugger via a Peripheral Component Interconnect (PCI) interface.

4. (Cancelled).

5. (Original) The method of claim 1, further comprising:

identifying within a first configuration file of a first debugger the first processing core associated with the first debugging session; and

identifying within a second configuration file of a second debugger the second processing core associated with the second debugging session.

6. (Original) The method of claim 5, further comprising:

routing, by the processor, the first debugger to the first processing core for establishing the first debugging session based on the first configuration file; and

routing, by the processor, the second debugger to the second processing core for establishing the second debugging session based on the second configuration file.

7. (Original) The method of claim 1, further comprising maintaining, by the processor, processor states while dynamically establishing the first and second debugging session.

8. (Currently Amended) A method, comprising:

receiving, by a processor, a first debugging session request initiated for a first application by a first device or first user;

receiving, by the processor, a second debugging session request initiated for a second application by a second device or a second user;

dynamically attaching a first debugger to a first processing core for servicing the first debugging session request; and

dynamically attaching a second debugger to a second processing core for servicing the second debugging request, and wherein the first debugging session operates independent of the second debugging session and both session proceed simultaneously within the processor on their respective processing cores and wherein the first application is different from the second application and the first device or the first user is different from the second device or second user, and wherein the processor is associated with a multi core processing architecture.

9. (Original) The method of claim 8, wherein dynamically attaching the first and second debuggers further includes identifying the first and second debuggers as a same debugger being initiated as independent and duplicative instances on different processing cores.

10. (Original) The method of claim 8, further comprising identifying within the first and the second debugging session requests configuration information which identifies the first and second processing cores.

11. (Original) The method of claim 8, wherein dynamically attaching the first and second debuggers further includes maintaining a previous state associated with the processor of the first and second processing cores before and after attaching the first and second debuggers to their respective processing cores.

12. (Original) The method of claim 8, wherein receiving the first and second debugging session requests further includes remotely initiating the requests from the processor that has the first and second processing cores.

13. (Original) The method of claim 8, further comprising maintaining existing states associated with existing applications, the existing applications processing on the first and second processing cores before and after dynamically attaching the first and second debuggers to the first and second processing cores, respectively.

14. (Original) The method of claim 8, wherein dynamically attaching the first and second debuggers further includes attaching the first and second debuggers to their respective processing cores as their respective processing cores are processing a number of other applications.

15. (Currently Amended) A system, comprising:

a processor having a first processing core and a second processing core, wherein the processor is associated with a multi core processor architecture; and

a debugger, wherein a first instance of the debugger is dynamically attachable to the first processing core of the processor and a second instance of the debugger is dynamically attachable to the second processing core of the processor, and wherein the first instance operates independent of the second instance and both instances simultaneously proceeds independent of the other instance, and wherein the first instance is associated with debugging a first application and the second instance is associated with debugging a second application and the first and second applications are different from one another.

16. (Original) The system of claim 15, further comprising a Peripheral Component Interconnect (PCI) interfaced to the processor for receiving requests to dynamically attach the first and second debugger instances to their respective processing cores.

17. (Original) The system of claim 15, further comprising a first configuration file associated with the first debugging instance and a second configuration file associated with the second debugging instance, wherein each configuration file identifies its respective processing core, and wherein the processor in response to the configuration files dynamically attaches the debugger instances to their respective processing cores.

18. (Cancelled).

19. (Original) The system of claim 15, wherein the processor maintains states associated with the first and second processing cores before and after the first and second instances are dynamically attached to their respective processing cores.

20. (Currently Amended) A machine accessible medium having associated instructions, which when accessed, results in a machine performing:

receiving a first request for a first debugging session to debug a first application and the first request received from a first device or first user;

receiving a second request for a second debugging session to debug a second application and the second application received from a second device or a second user, and wherein the first and second applications are different from one another and the first device or the first user are different from the second device or second user;

dynamically establishing the first debugging session on a first processing core of a processor associated with a multi core processing architecture; and

dynamically establishing the second debugging session on a second processing core of the processor, and wherein the first and second debugging sessions operate independent of one another on their respective processing cores and both sessions simultaneously proceed independent of one another.

21. (Cancelled).

22. (Original) The medium of claim 20, further including instructions for initiating a first debugger instance on the first processing core for managing the first debugging session and a second debugger instance on the second processing core for managing the second debugging session.

23. (Original) The medium of claim 20, further including instructions for maintaining states of the first and second processing cores before and after the first and second debugging sessions are dynamically established on their respective processing cores.

24. (Cancelled).

25. (Currently Amended) An apparatus, comprising:

configuration information associated with instances of a debugger; and

attachment logic residing within a processor having multiple processing cores and associated with a multi core processing architecture that dynamically attaches selective ones of the instances of the debuggers to selective ones of the processing cores in response to requests for debugging sessions having the configuration information, and wherein each debugger operates independent of remaining debuggers and each debugger debugs a different application from that which is associated with the remaining debuggers, and wherein each debugger simultaneously proceeds independent of the remaining debuggers.

26. (Original) The apparatus of claim 25, wherein the attachment logic maintains states of the processing cores before and after any dynamic attachment of the instances of the debugger.

27. (Original) The apparatus of claim 25, wherein configuration information is configurable parameter values provided with the requests for debugging sessions.

28. (Original) The apparatus of claim 25, wherein the configuration information is files, where each file is associated with a unique one of the debugging instances